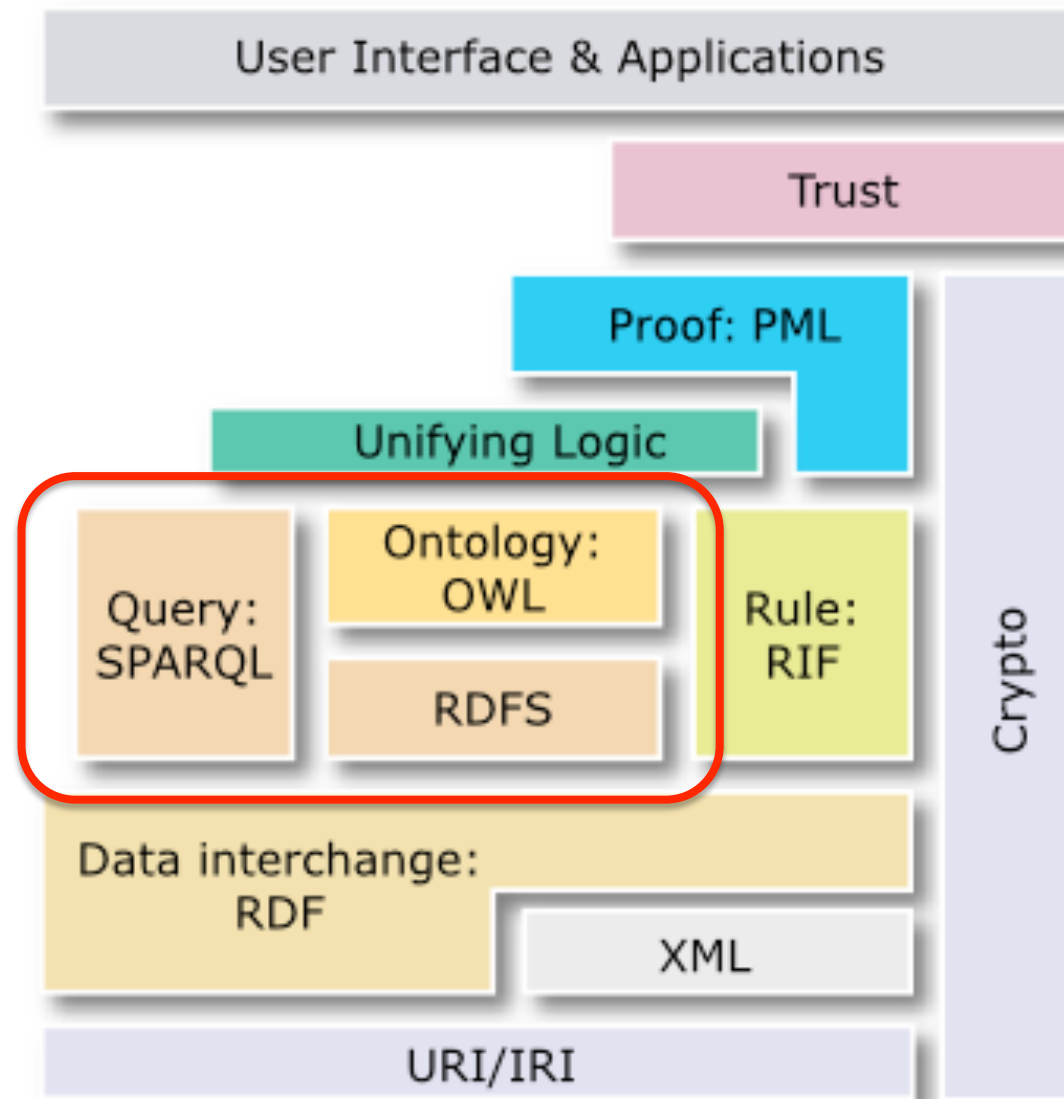


ESIP Semantic Web Tutorial – SPARQL Queries

ESIP 2011 Winter Meeting – Semantic Web Cluster
Thursday, January 6, 2011
Washington DC

Hook Hua

Semantic Web Stack

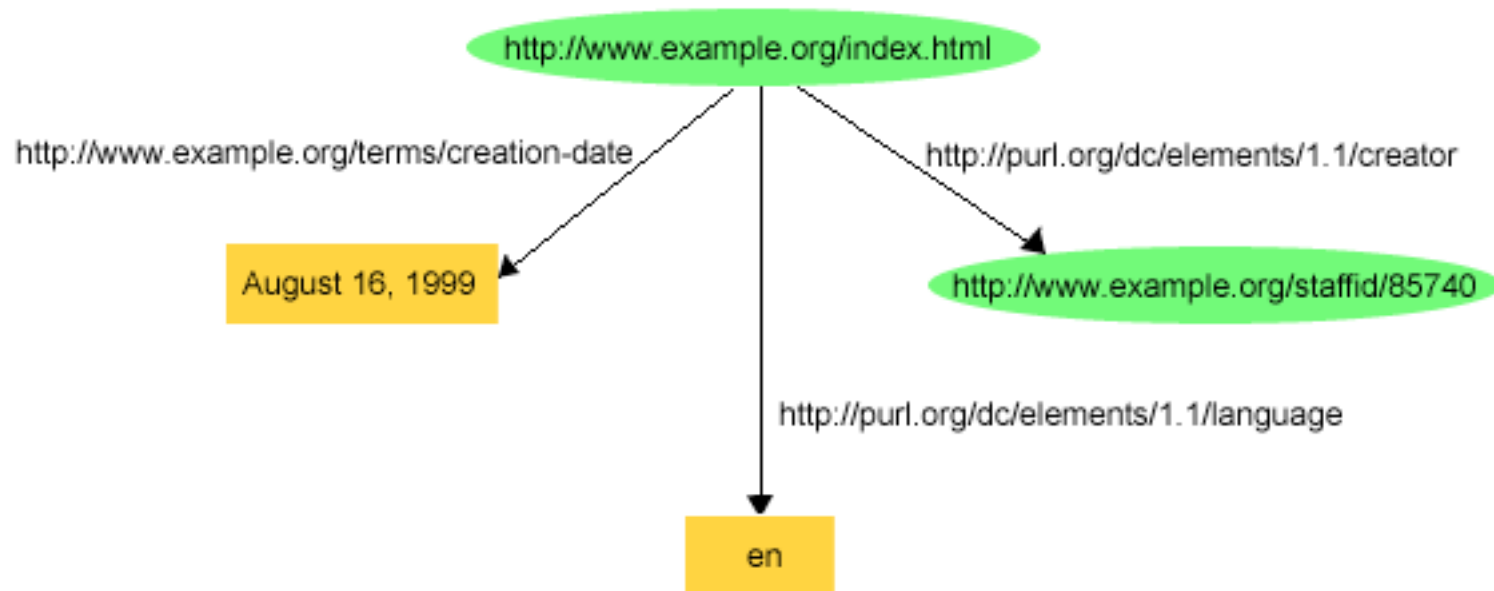


Resource Description Framework

RDF

Resource Description Framework (RDF)

- Built on the triple, a 3-tuple consisting of
 - Subject, Predicate, and Object
- Example graph below:
 - Resource: Some entity.
 - Property: An attribute of a resource.
 - Literal: A string of characters which can be the value of a property.



```
<http://www.example.org/index.html> <http://purl.org/dc/elements/1.1/creator> <http://www.example.org/staffid/85740> .  
<http://www.example.org/index.html> <http://www.example.org/terms/creation-date> "August 16, 1999" .  
<http://www.example.org/index.html> <http://purl.org/dc/elements/1.1/language> "en" .
```

RDF/XML Representation



- Long hand representation
 - URI references be written out completely, in angle brackets

```
<http://www.example.org/index.html> <http://purl.org/dc/elements/1.1/creator> <http://www.example.org/staffid/85740> .  
<http://www.example.org/index.html> <http://www.example.org/terms/creation-date> "August 16, 1999" .  
<http://www.example.org/index.html> <http://purl.org/dc/elements/1.1/language> "en" .
```

- Short hand representation
 - XML qualified name (or QName) without angle brackets
 - Uses form ***prefix:localname***

```
<?xml version="1.0"?>  
<rdf:RDF xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#  
  xmlns:dc=http://purl.org/dc/elements/1.1/  
  xmlns:ext=http://www.example.org/terms/>  
  <rdf:Description rdf:about="http://www.example.org/index.html">  
    <ext:creation-date>August 16, 1999</ext:creation-date>  
    <dc:language>en</dc:language>  
    <dc:creator rdf:resource="http://www.example.org/staffid/85740"/>  
  </rdf:Description>  
</rdf:RDF>
```

Common Namespace Prefix Bindings



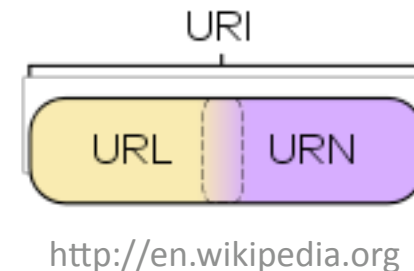
- Not hardwired, but commonly used prefix bindings

Prefix	IRI
rdf:	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs:	http://www.w3.org/2000/01/rdf-schema#
xsd:	http://www.w3.org/2001/XMLSchema#
fn:	http://www.w3.org/2005/xpath-functions#
foaf:	http://xmlns.com/foaf/0.1#
vcard	http://www.w3.org/2006/vcard/ns#
dc	http://purl.org/dc/elements/1.1/
opensearch	http://a9.com/-/spec/opensearch/1.1/
geo	http://www.w3.org/2003/01/geo/wgs84_pos#

“First Class Citizen” Resources



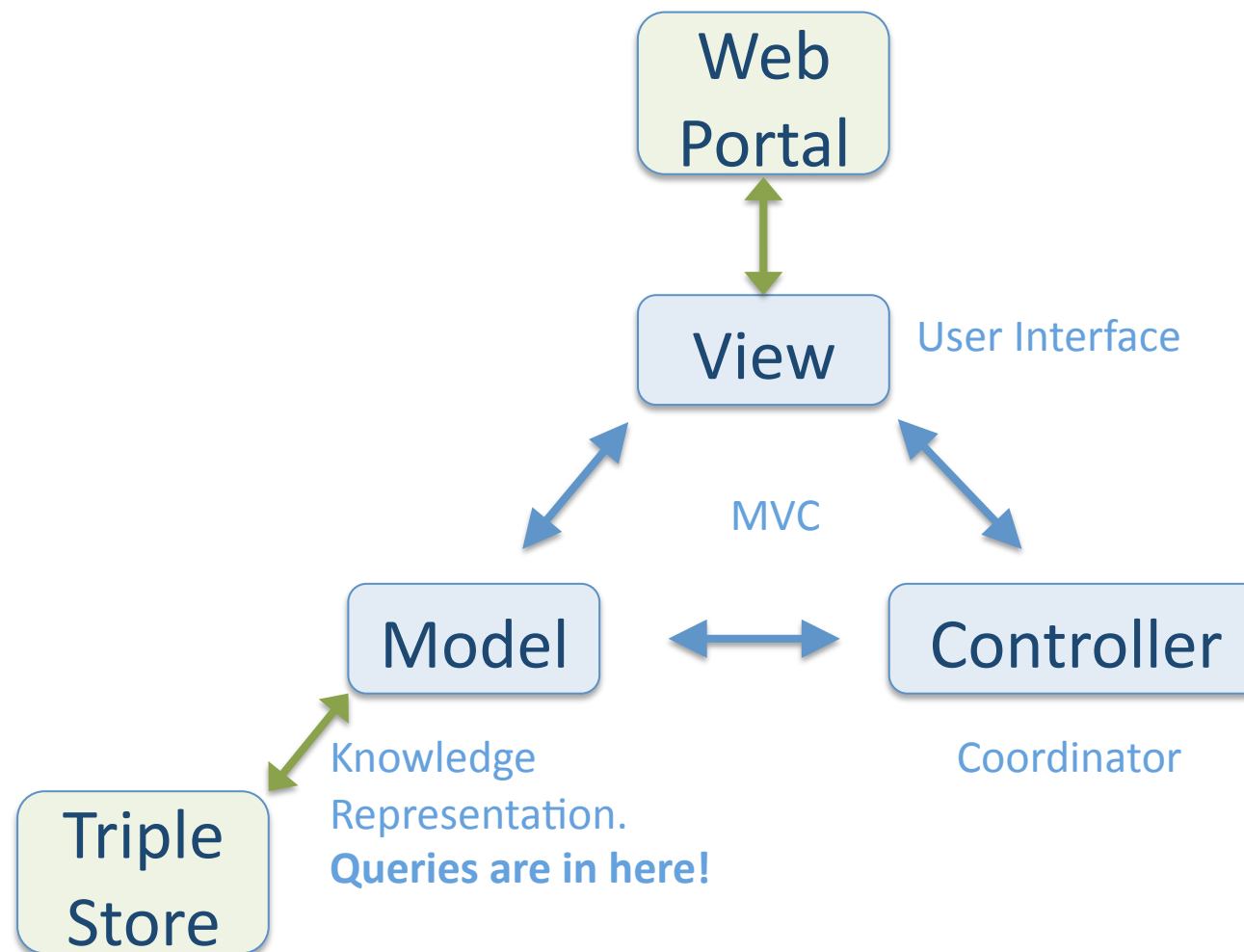
- Reification
 - Formulating a resource for conceptual manipulation
- Using formal Uniform Resource Identifiers (URIs)
 - identifying things using Web identifiers
 - <http://www.w3.org/TR/rdf-primer/>
 - *“RDF can also be used to represent information about things that can be identified on the Web, even when they cannot be directly retrieved on the Web.”*
 - Locators and Names
 - **<URI reference>#<URI fragment identifier>**
 - e.g. http://testbed.gmu.edu/esip/ESIP_Projects_v0.owl#worksWithTechnology



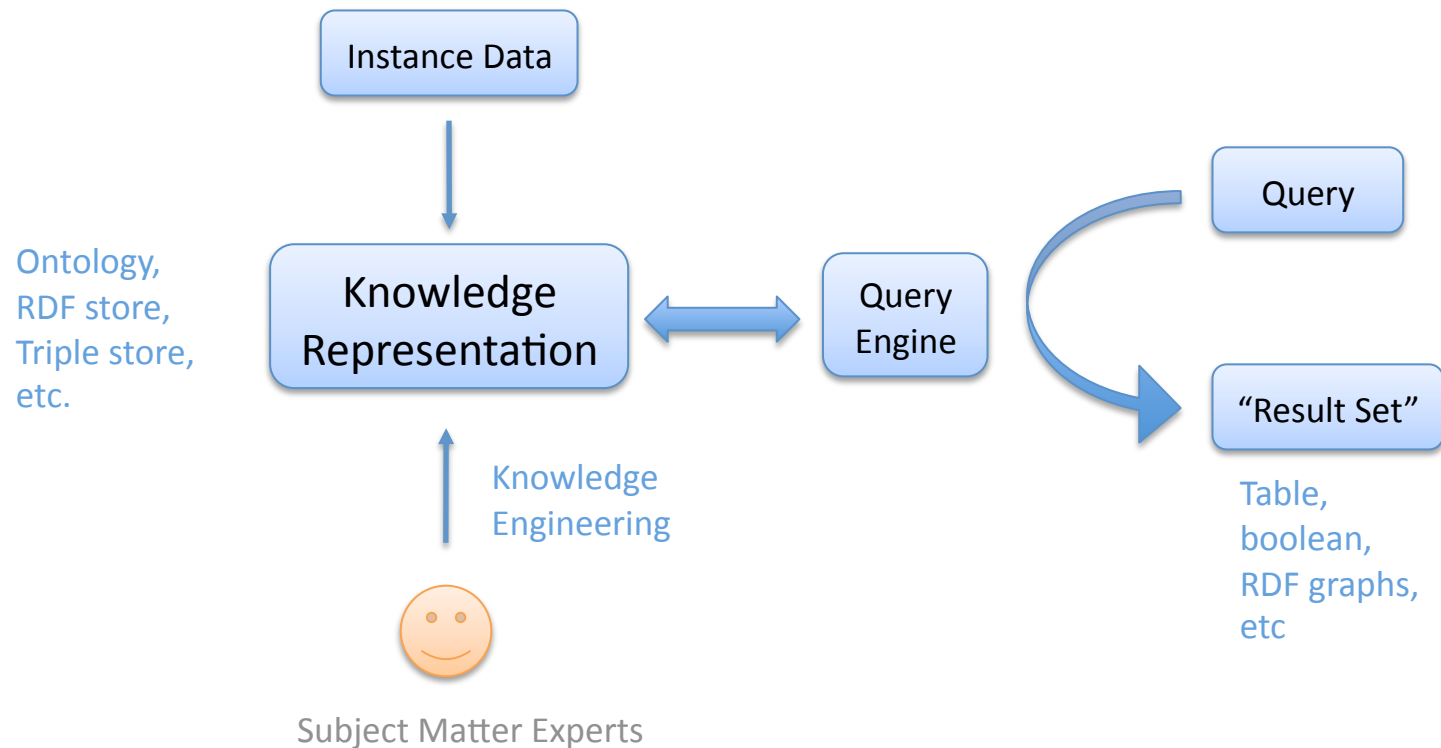
Looking up information

SPARQL QUERIES

How It Fits In (Example)



Applying Queries



Querying Languages

- SPARQL
 - W3C Standard query language for RDF
 - querying at RDF-level, not at OWL-level
- OWL query language (OWL-QL)
 - now obsolete
- SPARQUL
 - Peter Fox has been made a collaborator
- Query Lang
 - commercial and embedded QL
- SeRQL, RDFQuery, and many more...

What is SPARQL?

- Defined in terms of the W3C's RDF data model
 - Compatible with models that can be mapped into RDF
- Consists of three specifications
 1. Query language specification
 2. Query results XML format
 3. Data access protocol
- SPARQL is [currently] read-only
 - Does not have “CRUD” operators
- W3C Recommendation (2008-01-15)
 - <http://www.w3.org/TR/rdf-sparql-query/>

- ARQ
 - a SPARQL processor for Jena
- Pellet
 - OWL DL reasoner with some SPARQL query support
- Joseki
 - SPARQL server for Jena
 - HTTP implementation of SPARQL access protocol
 - <http://www.sparql.org/query.html>
- Rasqal
 - RDF query library in Redland framework
- Many others...

Structure of a SPARQL Query



- Simple example:

```
SELECT ?project ?technology
WHERE
{
  ?project <http://esipfed.orgworksWithTechnology> ?technology
}
```

- Add PREFIX namespace

```
PREFIX esip: <http://esipfed.org>
SELECT ?project ?technology
WHERE {
  ?project esip:worksWithTechnology ?technology
}
```

- **SELECT**
 - Return a table of results.
 - Results can be in text, XML, or even JSON
- **ASK**
 - Ask a boolean query.
 - Results can be in text, XML, or even JSON
- **CONSTRUCT**
 - Returns an RDF graph, based on a template in the query.
- **DESCRIBE**
 - Returns an RDF graph, based on what the query processor is configured to return.
- *Show Joseki's web SPARQL query form example..*
 - <http://www.sparql.org/query.html>

Testbed Hands-on



- Can follow along at
 - <http://testbed.gmu.edu/swtestbed/esipquery1.jsp>
- Testbed is work in progress.
 - Some “artifacts” only applicable to the current testbed

Note no “#” or “/” delimiter for loaded “esip” namespace

Note extra space at end of this line

```
PREFIX esip: <http://esipfed.org>
SELECT ?project ?technology
WHERE {
    ?project esip:worksWithTechnology ?technology
}
```


Let's See What's In the RDF Store

- SELECT *
- Not recommended for large triple stores

```
SELECT *  
WHERE {  
  ?s ?p ?o .  
}
```

- Lots of results

Note for current testbed store, results are out of order:
subject object predicate

```
http://www.w3.org/1999/02/22-rdf-syntax-ns#type http://www.w3.org/1999/02/22-rdf-syntax-ns#Property http://www.w3.org/1999/02/22-rdf-syntax-ns#type  
...  
http://www.w3.org/2000/01/rdf-schema#subPropertyOf http://www.w3.org/1999/02/22-rdf-syntax-ns#Property http://www.w3.org/1999/02/22-rdf-syntax-ns#type  
...  
http://tw.rpi.edu/person/PeterFox http://www.w3.org/2000/01/rdf-schema#Resource http://www.w3.org/1999/02/22-rdf-syntax-ns#type  
...  
http://testbed.gmu.edu/esip/ESIP_Projects_v0.owl#ProjectParticipant http://www.w3.org/2000/01/rdf-schema#Resource http://www.w3.org/1999/02/22-rdf-syntax-ns#type  
...  
http://tw.rpi.edu/project/DQSS http://www.w3.org/2000/01/rdf-schema#Resource http://www.w3.org/1999/02/22-rdf-syntax-ns#type  
...  
http://tw.rpi.edu/project/DQSS "NASA ACCESS" http://testbed.gmu.edu/esip/ESIP_Projects_v0.owl#isFundedBy  
...  
http://testbed.gmu.edu/esip/ESIP_Projects_v0.owl#worksWithTechnology http://testbed.gmu.edu/esip/ESIP_Projects_v0.owl#worksWithTechnology http://www.w3.org/2000/01/rdf-  
schema#subPropertyOf  
...  
http://tw.rpi.edu/project/GeossClearinghosue "GeoNetwork" http://esipfed.orgworksWithTechnology
```

Filtering Solutions/ regex()

- FILTERs restrict solutions to those for which the *filter expression* evaluates to TRUE.
- Sometimes do not know the exact resource names
- Use regex() for partial matches
 - Tests whether a literal value contains a certain substring
 - Case sensitive regex(?x, "Foo')
 - Case insensitive regex(?x, "foo', "I")

```
SELECT ?s ?p ?o
WHERE {
  ?s ?p ?o .
  FILTER (regex(?o, "ACCESS", "i")) .
}
```

*Find all statements
with "ACCESS" as the
object value*

- regex()'s first argument has to be either a **plain literal** without a language tag or a typed literal with a datatype of xsd:string.
- Otherwise, can using **str()** function to convert a resource plain literal without language tag

```
SELECT ?p
WHERE {
  ?s ?p ?o .
  FILTER (regex(str(?p), "technology", "i")) .
}
```

*Find all
statements that
contain the
substring
"technology"*

Example Literals



- "chat"
- 'chat'@en with language tag "en"
- "xyz"^^http://example.org/ns/userDatatype
- "abc"^^appNS:appDataType
- "'The librarian said, 'Perhaps you would enjoy 'War and Peace'.''"
- 1, which is the same as "1"^^xsd:integer
- 1.3, which is the same as "1.3"^^xsd:decimal
- 1.0e6, which is the same as "1.0e6"^^xsd:double
- true, which is the same as "true"^^xsd:boolean

<http://www.w3.org/TR/rdf-sparql-query/>

Solution Modifiers

- ORDER
 - put the solutions in order
- PROJECTION
 - choose certain variables
- DISTINCT
 - ensure solutions in the sequence are unique
- REDUCE
 - permit elimination of some non-unique solutions
- OFFSET
 - control where the solutions start from in the overall sequence of solutions
- LIMIT
 - restrict the number of solutions

<http://www.w3.org/TR/rdf-sparql-query/>

Modifier: ORDER BY

- **ORDER BY** sorts the results
- Sequence of order comparators is composed of an expression and an optional order modifier (either **ASC()** or **DESC()**).
- Note that it uses the “<” operator for resource comparisons.

```
SELECT ?p
WHERE {
  ?s ?p ?o .
  FILTER (regex(str(?p), "technology", "i")) .
}
```

Find all predicates containing the substring “technology”.

```
SELECT ?p
WHERE {
  ?s ?p ?o .
  FILTER (regex(str(?p), "technology", "i")) .
}
ORDER BY DESC(?p)
```

Find all predicates containing the substring “technology”. Sort results in descending order.

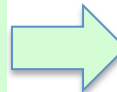
Modifier: DISTINCT

- **DISTINCT** eliminates duplicate solutions

```
SELECT DISTINCT ?p
WHERE {
  ?s ?p ?o .
  FILTER (regex(str(?p), "technology", "i")) .
}
ORDER BY DESC(?p)
```

*Find all predicates
containing the
substring
"technology".
Eliminate
duplicates.*

http://www.w3.org/2001/XMLSchema#worksWithTechnology
http://www.w3.org/2001/XMLSchema#worksWithTechnology
http://www.w3.org/2001/XMLSchema#worksWithTechnology
http://testbed.gmu.edu/esip/ESIP_Projects_v0.owl#worksWithTechnology
http://testbed.gmu.edu/esip/ESIP_Projects_v0.owl#worksWithTechnology
http://esipfed.orgworksWithTechnology
http://esipfed.orgworksWithTechnology
http://esipfed.orgworksWithTechnology
http://esipfed.orgworksWithTechnology
http://esipfed.orgworksWithTechnology
http://esipfed.orgworksWithTechnology
http://esipfed.orgTechnology
http://esipfed.orgTechnology



http://www.w3.org/2001/XMLSchema#worksWithTechnology
http://testbed.gmu.edu/esip/ESIP_Projects_v0.owl#worksWithTechnology
http://esipfed.orgworksWithTechnology
http://esipfed.orgTechnology

- UNION
 - combine graph patterns so that one of several alternative graph patterns may match

```
PREFIX esip: <http://esipfed.org>
SELECT ?project
WHERE {
  { ?project esip:isFundedBy "NASA ACCESS" }
  UNION
  { ?project esip:worksWithTechnology "GeoNetwork" }
}
```

*Find all projects
that are either
funded by ACCESS
or works with
GeoNetwork
technology.*

- **OFFSET**

- Start solutions at the given offset index

```
SELECT DISTINCT ?s
WHERE {
    ?s ?p ?o .
}
ORDER BY ASC(?s)
OFFSET 0
LIMIT 5
```

*Find all distinct subjects, but only show **first** page with 5 results per page.*

- **LIMIT**

- Upper bound of number of solutions

```
SELECT DISTINCT ?s
WHERE {
    ?s ?p ?o .
}
ORDER BY ASC(?s)
OFFSET 5
LIMIT 5
```

*Find all distinct subjects, but only show **second** page with 5 results per page.*

- *Use together for pagination!*

- What projects are working with “Jena” (a semantic web technology)?

Exploratory query

```
SELECT ?s ?p ?o
WHERE {
  ?s ?p ?o
  FILTER (regex(str(?o), "Jena", "i")) .
}
```

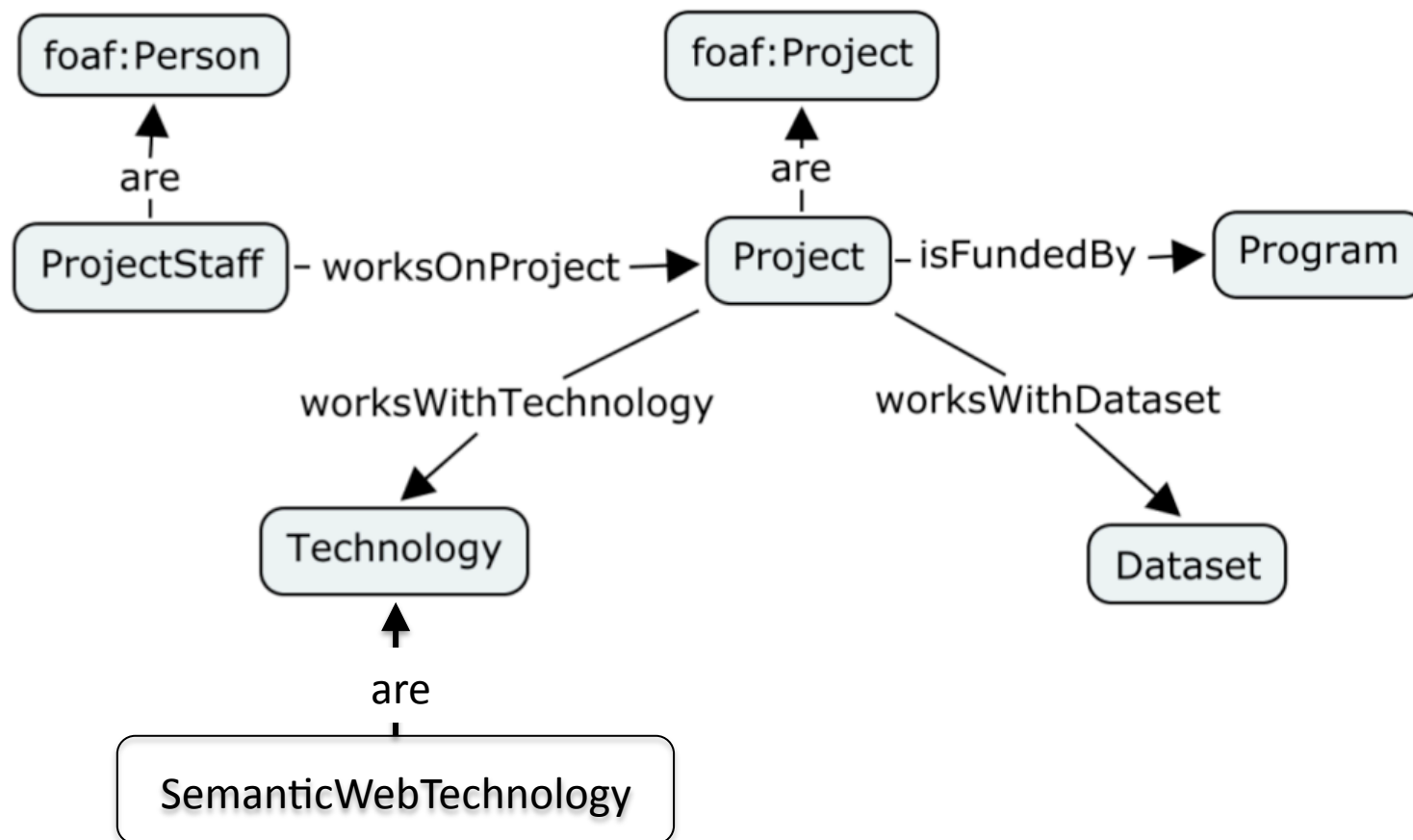
Refined query

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX esip: <http://esipfed.org>
PREFIX testbed: <http://testbed.gmu.edu/esip/ESIP_Projects_v0.owl#>

SELECT ?project ?technology
WHERE {
  ?project rdf:type esip:Project .
  ?project esip:worksWithTechnology ?technology .
  FILTER (regex(str(?technology), "Jena", "i")) .
}
```

“a semantic web technology”

- Options to better classify types of technologies
 - Add new technology type attribute to Technology class. Run classifier later.
 - Subclass Technology with more specific SemanticWebTechnology:



Use Cases



- What technologies are used by NASA's ACCESS program?

Exploratory query

```
SELECT ?project ?tech
WHERE {
  ?project ?p ?access .
  FILTER (regex(str(?access), "access", "i")) .

  ?project ?worksWith ?tech .
  FILTER (regex(str(?worksWith), "workswith", "i")) .
}
```

Refined query

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX esip: <http://esipfed.org>
PREFIX testbed: <http://testbed.gmu.edu/esip/ESIP_Projects_v0.owl#>

SELECT DISTINCT ?technology
WHERE {
  ?project rdf:type esip:Project .
  ?project esip:worksWithTechnology ?technology .
  ?project testbed:isFundedBy ?access .
  FILTER (regex(str(?access), "access", "i")) .
}
```

Basic Federated SPARQL Query



- SERVICE extension can query remote RDF endpoints
- Jena ARQ supports extension (if the query is parsed with language Syntax.syntaxARQ)

```
PREFIX : http://example/  
PREFIX dc: http://purl.org/dc/elements/1.1/  
SELECT ?a  
FROM <mybooks.rdf> {  
  ?b dc:title ?title .  
  SERVICE http://sparql.org/books {  
    ?s dc:title ?title . ?s dc:creator ?a  
  }  
}
```

Find the creators of books in mybooks.rdf. Cross-references book titles with <http://sparql.org/books> which contains the creator information

Do not expose underlying querying language to users

Do not expect users to know your ontology terms exactly

RDFa

RESOURCE DESCRIPTION FRAMEWORK – IN – ATTRIBUTES

Resource Description Framework – in – attributes (RDFa)



- W3C Recommendation
 - <http://www.w3.org/2006/07/SWD/RDFa/syntax/>
- Provides a set of attributes that can be used to carry metadata in XHTML
- Attributes

about and **src** – a URI or [CURIE](#) specifying the resource the metadata is about
rel and **rev** – specifying a relationship or reverse-relationship with another resource
href and **resource** – specifying the partner resource
property – specifying a property for the content of an element
content – optional attribute that overrides the content of the element when using the property attribute
datatype – optional attribute that specifies the datatype of text specified for use with the property attribute
typeof – optional attribute that specifies the RDF type(s) of the subject (the resource that the metadata is about).

- Example HTML snippet

```
<p> My <span rel="foaf:interest" resource="urn:ISBN:0752820907">favorite book is  
the inspiring <span about="urn:ISBN:0752820907"><cite property="dc:title">Weaving  
the Web</cite> by <span property="dc:creator">Tim Berners-Lee</span></span  
</span> </p>
```

- XHTML chunk

This photo was taken by `Mark Birbeck`.

- Simple attribute augmentation

This photo was taken by
`Mark Birbeck`.

- Yields RDF triple

`<photo1.jpg> dc:creator "Mark Birbeck"^^rdf:XMLLiteral .`